FunG - Biomechanical Models in Modern C++

Lars Lubkoll
Yuan-Cheng Fung

- Founder of “Modern” Biomechanics
Yuan-Cheng Fung

- Founder of “Modern” Biomechanics
- Fung’s law: $\frac{\partial \sigma}{\partial C} = a \sigma$
Half a model for muscle tissue

\[ W(F) = c \left[ \exp(b(\bar{\iota}_1(C) - 3)) - 1 \right] + d \left[ \exp(a(\bar{\iota}_6(C, M) - 1)^2) - 1 \right] \]

\[ \bar{\iota}_1 = \text{tr}(C) \det(C)^{-1/3}, \quad C = F^T F \]

\[ \bar{\iota}_6 = \text{tr}(CM^2) \det(C)^{-1/3} \]

\( F \): deformation gradient
\( M \): structural tensor
Half a model for muscle tissue

\[\text{auto model} = c \ast ( \exp(b \ast (m_{i1}(F)-3)) - 1 ) + d \ast ( \exp(a \ast (m_{i6}(F,M)-1)^2) - 1 );\]
\[\text{auto W} = \text{finalize} (\text{model} (\text{strainTensor}(F)) );\]

**F**: deformation gradient  
**M**: structural tensor
Usage (given matrices, F, dFi):

// set function argument
W.update(F);

// access function value and derivatives
auto value = W(); // or W.d0()
auto df1 = W.d1(dF0);
auto df2 = W.d2(dF0, dF1);
auto df3 = W.d3(dF0, dF1, dF2);
Techniques
Expression Templates + Operator Overloading
Expression Templates + Operator Overloading

- template programming (almost) without templates

```cpp
template <class Arg, int n = dim<Arg>()>
auto mi1(const Arg& x)
{
    return i1(x)*pow<-1,n>(det(x));
}
```
Expression Templates + Operator Overloading

- template programming (almost) without templates
- enhance inlining capabilities

```cpp
template <class Arg, int n = dim<Arg>()>
auto mil(const Arg& x) {
    return i1(x)*pow<-1,n>(det(x));
}
```
Generic Programming
Generic Programming

- no types:
  - `auto`, `decltype`, `std::declval`
Generic Programming

- no types:
  → auto, decltype, std::declval

- (relaxed) vector space structure for user-defined types
Generic Programming

- no types:  
  → auto, decltype, std::declval
- (relaxed) vector space structure for user-defined types
- SFINAE-adjustments for Armadillo, Eigen, Dune::FieldMatrix
Optimization
Caching
Elimination of zeros \( \nu_4(F^T F, M) = \text{tr}(f), f = F^T FM \)
Speak with the compiler:

- max-inline-insns-auto = 5000
- early-inlining-insns = 5000
- inline-unit-growth = 100

Factor: up to 14x (av.: 2.5x)
Speak with the compiler:

- `max-inline-insns-auto = 5000`
Speak with the compiler:

- `max-inline-insns-auto = 5000`
- `early-inlining-insns = 5000`

Weg wird bei Schnee - u. Eisglätte

nicht gerämt und gestreut.

Away at the Snow and Ice

is not cleared and Dispersed day
Speak with the compiler:

- max-inline-insns-auto = 5000
- early-inlining-insns = 5000
- inline-unit-growth = 100
  
  Factor: up to 14x (av.: 2.5x)
Performance
Comparison with AD libraries

\[ f(x) = x^{3/2} + \sin(\sqrt{x}) \quad (10^7 \text{ eval.}) \]

- **FADBAD++**: 0.58 time/s
- **SACADO**: 0.54 time/s
- **FunG**: 0.52 time/s
- **MANUAL**: 0.51 time/s
Comparison with AD libraries

\[ f(x, y, z) = (y + z)\sqrt{x} + \sin(\sqrt{x}) \quad (10^7 \text{ eval.}) \]
Reliability

• unit and integration tests
• compile-time concept checks
Reliability

- unit and integration tests
Reliability

- unit and integration tests
- compile-time concept checks
Half a model for muscle tissue

\[ W(F) = c \left[ \exp(b(\bar{i}_1 - 3)) - 1 \right] + d \left[ \exp(a(\bar{i}_6 - 1)^2) - 1 \right] \]

(10^7 eval.)